



# TP de Especificación

## Trabajo Práctico Grupal

30 de Marzo de 2022

Algoritmos y Estructuras de Datos I

### Grupo 10

Integrante	LU	Correo electrónico
Dominguez, Emilia	37752993	maemiliadominguez@gmail.com
Kerbs, Octavio	64/22	octaviokerbs@gmail.com
Russo, Gabriel	107/19	gabrielrussoguiot@gmail.com
Traverso, Lucas	479/18	lucas6246@gmail.com



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (+54 +11) 4576-3300

<http://www.exactas.uba.ar>

# Índice

<b>1. Definición de Tipos</b>	<b>2</b>
<b>2. Problemas</b>	<b>3</b>
2.1. Parte I: Juego básico . . . . .	3
2.1.1. Ejercicio 1 . . . . .	3
2.1.2. Ejercicio 2 . . . . .	3
2.1.3. Ejercicio 3 . . . . .	3
2.1.4. Ejercicio 4 . . . . .	3
2.1.5. Ejercicio 5 . . . . .	3
2.1.6. Ejercicio 6 . . . . .	4
2.2. Parte II: Despejar los vacíos . . . . .	5
2.2.1. Ejercicio 7 . . . . .	5
2.2.2. Ejercicio 8 . . . . .	5
2.3. Parte III: Jugador automático . . . . .	6
2.3.1. Ejercicio 9 . . . . .	6
<b>3. Funciones Auxiliares Y Predicados</b>	<b>7</b>
3.1. Ejercicio 1 . . . . .	7
3.2. Ejercicio 2 . . . . .	7
3.3. Ejercicio 3 . . . . .	7
3.4. Ejercicio 4 . . . . .	8
3.5. Ejercicio 5 . . . . .	8
3.6. Ejercicio 6 . . . . .	8
3.7. Ejercicio 7 . . . . .	8
3.8. Ejercicio 8 . . . . .	9
3.9. Ejercicio 9 . . . . .	10

## 1. Definición de Tipos

type  $pos = \mathbb{Z} \times \mathbb{Z}$

type  $tablero = seq\langle seq\langle Bool \rangle \rangle$

type  $jugadas = seq\langle pos \times \mathbb{Z} \rangle$

type  $banderitas = seq\langle pos \rangle$

## 2. Problemas

### 2.1. Parte I: Juego básico

#### 2.1.1. Ejercicio 1

$\text{aux minasAdyacentes } (t: \text{tablero}, p: \text{pos}) : \mathbb{Z} =$   
 $\sum_{i=\max(p[0]-1, 0)}^{\min(p[0]+1, |t|-1)} \sum_{j=\max(p[1]-1, 0)}^{\min(p[1]+1, |t|-1)} \text{es1SiPosicionEsBombaSino0}(t, i, j) -$   
 $\text{es1SiPosicionEsBombaSino0}(t, p[0], p[1]);$

#### 2.1.2. Ejercicio 2

$\text{pred juegoValido } (t: \text{tablero}, j: \text{jugadas}) \{$   
     $(\text{tableroValido}(t) \wedge$   
     $\text{todasLasPosicionesDeLaJugadaPertenecenAlTablero}(t, j) \wedge$   
     $\text{noExistenPosicionesRepetidasEnLaJugada}(j)) \wedge_L$   
     $(\text{esLaCantidadDeMinasAdyacentesCorrectaParaTodaLaJugada}(t, j) \wedge$   
     $\text{cantidadDeBombasEnPosicionesDeLaJugada}(t, j) \leq 1)$   
 $\}$

#### 2.1.3. Ejercicio 3

$\text{proc plantarBanderita } (\text{in } t: \text{tablero}, \text{in } j: \text{jugadas}, \text{in } p: \text{pos}, \text{inout } b: \text{banderitas}) \{$   
     $\text{Pre } \{ \text{juegoValido}(t, j) \wedge$   
     $\text{posicionPerteneceAlTablero}(t, p) \wedge$   
     $\neg \text{posicionPerteneceAJugada}(p, j) \wedge$   
     $\neg \text{posicionPerteneceABanderitas}(p, b) \wedge$   
     $\text{banderitasValidasParaLaJugada}(b, j, t) \wedge$   
     $b = b_0 \}$   
     $\text{Post } \{ \text{posicionPerteneceABanderitas}(p, b) \wedge$   
     $\text{todasLasPosicionesDeB}_1 \text{PertenecenAB}_2(b_0, b) \wedge$   
     $(|b| = |b_0| + 1) \}$   
 $\}$

#### 2.1.4. Ejercicio 4

$\text{proc perdió } (\text{in } t: \text{tablero}, \text{in } j: \text{jugadas}, \text{out } \text{res}: \text{Bool}) \{$   
     $\text{Pre } \{ \text{juegoValido}(t, j) \}$   
     $\text{Post } \{ \text{res} = \text{true} \iff \text{cantidadDeBombasEnPosicionesDeLaJugada}(t, j) = 1 \}$   
 $\}$

#### 2.1.5. Ejercicio 5

$\text{proc ganó } (\text{in } t: \text{tablero}, \text{in } j: \text{jugadas}, \text{out } \text{res}: \text{Bool}) \{$   
     $\text{Pre } \{ \text{juegoValido}(t, j) \}$   
     $\text{Post } \{ \text{res} = \text{true} \iff \text{cantidadDeBombasEnPosicionesDeLaJugada}(t, j) = 0 \wedge$   
     $\text{jugadasTodasLasPosicionesSinBombas}(t, j) \}$   
 $\}$

### 2.1.6. Ejercicio 6

```
proc jugar (in t: tablero , in b: banderitas, in p: pos, inout j: jugadas) {  
  Pre {juegoValido(t, j)  $\wedge$   
    posicionPerteneceAlTablero(t, p)  $\wedge$   
     $\neg$  posicionPerteneceAJugada(p, j)  $\wedge$   
     $\neg$  posicionPerteneceABanderitas(p, b)  $\wedge$   
    banderitasValidasParaLaJugada(b, j, t)  $\wedge_L$   
    (juegoEnMarcha(j, t)  $\wedge$   
    j = j0)}  
  Post {posicionPerteneceAJugada(p, j)  $\wedge$   
    todasLasJ1PertenecenAJ2(j0, j)  $\wedge$   
    (|j| = |j0| + 1)}  
}
```

## 2.2. Parte II: Despejar los vacíos

### 2.2.1. Ejercicio 7

```
pred caminoLibre (t: tablero, p0: pos, p1: pos) {  
  (∃s : seq < pos >)(caminoLibreSinDefinirMinasAdyacentesALaUltimaPosicion(t, p0, p1, s) ∧  
    8 > minasAdyacentes(t, p1) ≥ 1)  
}
```

### 2.2.2. Ejercicio 8

```
proc jugarPlus (in t: tablero , in b: banderitas, in p: pos, inout j: jugadas) {  
  Pre {juegoValido(t, j) ∧  
    posicionPerteneceAlTablero(t, p) ∧  
    ¬ posicionPerteneceAJugada(p, j) ∧  
    ¬ posicionPerteneceABanderitas(p, b) ∧  
    banderitasValidasParaLaJugada(b, j, t) ∧  
    (juegoEnMarcha(j, t) ∧  
    jugadasExtendidasValidas(t, j, b) ∧  
    j = j0)}  
  Post {(posicionPerteneceAJugada(p, j) ∧  
    todasLasJ1PertenecenAJ2(j0, j) ∧  
    juegoValido(t, j)) ∧  
    (jugadasExtendidasValidas(t, j, b) ∧  
    (∀q : pos)(posicionPerteneceAlTablero(t, q) ∧  
    ¬ posicionPerteneceAJugada(q, j0) ∧  
    q ≠ p ∧  
    ¬ (∃s : seq<pos>)(caminoLibreSinDefinirMinasAdyacentesALaUltimaPosicionConBanderitas(t, p, q, s, b)) —  
    ¬ posicionPerteneceAJugada(q, j)))}  
}
```

## 2.3. Parte III: Jugador automático

### 2.3.1. Ejercicio 9

```
proc sugerirAutomático121 (in t: tablero , in b: banderitas, out p: pos) {  
  Pre {juegoValido(t, j)  $\wedge$  juegoEnMarcha(j, t)  $\wedge$   
    hayPatron121(t, j)}  
  
  Post {posicionPerteneceAlTablero(p, t)  $\wedge_L$   
    posicionNoPerteneceAJugadas(p, j)  $\wedge$   
    ( $\exists s : seq\langle pos \rangle$ )(esPatron121(t, j, s)  $\wedge_L$  sonAdyacentesNoDiagonales(p, s[1]))}  
}
```

### 3. Funciones Auxiliares Y Predicados

#### 3.1. Ejercicio 1

```
aux es1SiPosicionEsBombaSino0 (t: tablero, x, y:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  = if(t[x][y] = true) then 1 else 0 fi;;  
aux max (x,y:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  = if(x < y) then y else x;  
aux min (x,y:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  = if(x < y) then x else y;
```

#### 3.2. Ejercicio 2

```
pred tableroValido (t: tablero) {  
    ( $\forall i : \mathbb{Z}$ )( $0 \leq i < |t| \rightarrow_L |t| = |t[i]|$ )  $\wedge_L$  (cantidadTotalDeBombas(t) > 0)  
}  
  
aux cantidadTotalDeBombas (t: tablero) :  $\mathbb{Z}$  =  $\sum_{i=0}^{|t|-1} \sum_{j=0}^{|t[i]|-1} es1SiPosicionEsBombaSino0(t, i, j)$ ;  
  
pred todasLasPosicionesDeLaJugadaPertenecenAlTablero (t: tablero, j: jugada) {  
    ( $\forall i : \mathbb{Z}$ )( $0 \leq i < |j| \rightarrow_L posicionPerteneceAlTablero(t, j[i][0])$ )  
}  
  
pred posicionPerteneceAlTablero (p: pos, t: tablero) {  
    ( $0 \leq p_0 < |t|$ )  $\wedge$  ( $0 \leq p_1 < |t|$ )  
}  
  
pred noExistenPosicionesRepetidasEnLaJugada (j: jugadas) {  
    ( $\forall i : \mathbb{Z}$ )( $\forall k : \mathbb{Z}$ )( $(0 \leq i < |j| \wedge 0 \leq k < |j| \wedge i \neq k) \rightarrow_L j[i][0] \neq j[k][0]$ )  
}  
  
pred esLaCantidadDeMinasAdyacentesCorrectaParaTodaLaJugada (t: tablero, j: jugadas) {  
    ( $\forall i : \mathbb{Z}$ )( $0 \leq i < |j| \rightarrow_L j[i][1] = minasAdyacentes(t, j[i][0])$ )  
}  
  
3.3. Ejercicio 3  
  
pred posicionPerteneceAJugada (p: pos, j: jugada) {  
    ( $\exists i : \mathbb{Z}$ )( $0 \leq i < |j| \wedge_L j[i][0] = p$ )  
}  
  
pred posicionPerteneceABanderitas (p: posicion, b: banderita) {  
    ( $\exists i : \mathbb{Z}$ )( $0 \leq i < |b| \wedge_L b[i] = p$ )  
}  
  
pred banderitasValidasParaLaJugada (b: banderitas, j: jugadas, t: tablero) {  
    noExistenPosicionesRepetidasEnLasBanderitas(b)  $\wedge$   
    todasLasPosicionesDeLasBanderitasPertenecenAlTablero(t, b)  $\wedge$   
    ningunaPosicionDeLaJugadaEstaEnBanderitas(j, b)  
}
```



```

pred noExistenPosicionesRepetidasEnLasBanderitas (b: banderitas) {
  (∀i:ℤ)(∀k:ℤ)((0 ≤ i < |b| ∧ 0 ≤ k < |b| ∧ i ≠ k)
    →L b[i] ≠ b[k])
}

pred todasLasPosicionesDeLasBanderitasPerteneceAlTablero (t: tablero, b: banderitas) {
  (∀i:ℤ)(0 ≤ i < |b| →L posicionPerteneceAlTablero(t, b[i]))
}

pred ningunaPosicionDeLaJugadaEstaEnBanderitas (j: jugadas, b: banderitas) {
  (∀i:ℤ)(∀k:ℤ)((0 ≤ i < |j| ∧ 0 ≤ k < |b|) →L j[i][0] ≠ b[k])
}

pred todasLasPosicionesDeB1PerteneceAB2 (b1, b2: banderitas) {
  (∀x:pos)(posicionPerteneceABanderitas(x, b1) → posicionPerteneceABanderitas(x, b2))
}

```

### 3.4. Ejercicio 4

aux cantidadDeBombasEnPosicionesDeLaJugada (t: *tablero*, j: *jugada*) : ℤ =  
 $\sum_{i=0}^{|j|-1} \text{es1SiPosicionEsBombaSino0}(t, j[i][0][0], j[i][0][1]);$

### 3.5. Ejercicio 5

```

pred jugadasTodasLasPosicionesSinBombas (t: tablero, j: jugadas) {
  |j| = posicionesSinMinas(t)
}

aux posicionesSinMinas (t: tablero) : ℤ = (∑i=0|t|-1 ∑k=0|t|-1 if(t[i][k] = false) then 1 else 0 fi);

```

### 3.6. Ejercicio 6

```

pred juegoEnMarcha (j: jugadas, t: tablero) {
  cantidadDeBombasEnPosicionesDeLaJugada(t, j) = 0 ∧ ¬ jugadasTodasLasPosicionesSinBombas(t, j)
}

pred todasLasJ1PerteneceAJ2 (j1, j2: jugadas) {
  (∀x:pos)(posicionPerteneceAJugada(x, j1) → posicionPerteneceAJugada(x, j2))
}

```

### 3.7. Ejercicio 7

```

pred caminoLibreSinDefinirMinasAdyacentesALaUltimaPosicionConBanderitas (t: tablero, p0: pos, p1: pos, s: seq⟨pos⟩,
b: banderitas) {
  (posicionPerteneceASecuencia(p0, s) ∧
  posicionPerteneceASecuencia(p1, s) ∧
  (∀p:pos)(posicionPerteneceASecuencia(p, s) → posicionPerteneceAlTablero(t, p))) ∧L
  ((∀p:pos)(posicionPerteneceASecuencia(p, s) ∧ p ≠ p1 → minasAdyacentes(t, p) = 0) ∧
  (∃s2: seq < pos >)(ningunaPosicionDeLaSecuenciaEstaEnBanderitas(s2, b) ∧
  secuenciaDePosicionesAdyacentes(p0, p1, s2) ∧ esPermutacion(s, s2)))
}

```

```

pred ningunaPosicionDeLaSecuenciaEstaEnBanderitas (s: seq⟨pos⟩, b: banderitas) {
  (∀i:ℤ)(∀k:ℤ)((0 ≤ i < |b| ∧ 0 ≤ k < |s|) →L s[k] ≠ b[i])
}

```

```

pred posicionPerteneceASecuencia (p: pos, s: seq⟨pos⟩) {
  (∃i:ℤ)(0 ≤ i < |s| ∧L s[i] = p)
}

```

```

pred secuenciaDePosicionesAdyacentes (p1, p2: pos, s: seq⟨pos⟩) {
  (s[0] = p[1] ∧ s[|s| - 1] = p[2]) ∧
  (∀i:ℤ)(0 ≤ i < |s| - 1 →L esAdyacente(p[i], p[i + 1]))
}

```

### 3.8. Ejercicio 8

```

pred jugadasExtendidasValidas (t: tablero, j: jugadas, b: banderitas) {
  (∀p: pos)(posicionPerteneceAJugada(p, j) →L
  ((0 = minasAdyacentes(t, p) ∧
  todasLasPosicionesConMinasAdyacentesYCaminoLibreAPertenecenAJugada(p, j, b)) ∨
  (0 ≠ minasAdyacentes(t, p) ∧
  todasLasPosicionesSinMinasAdyacentesConUnaPosicionConfirmadaEnJugadaYCaminoLibreAPertenecenAJugada(p,
  j)
  )
  )
}

```

```

pred todasLasPosicionesConMinasAdyacentesYCaminoLibreAPertenecenAJugada (t: tablero, p: posicion, j: jugada, b:
banderitas) {
  (∀q: pos)((posicionPerteneceAlTablero(t, q) ∧
  ¬ posicionPerteneceABanderitas(q, b)) ∧L
  (0 ≠ minasAdyacentes(t, q) ∧
  caminoLibreConBanderitas(t, p, q, b))) →L
  posicionPerteneceAJugada(q, j))
}

```

```

pred esAdyacente (p, q: pos) {
  q[0] - 1 ≤ p[0] ≤ q[0] + 1 ∧ q[1] - 1 ≤ p[1] ≤ q[1] + 1 ∧ p ≠ q
}

```

```

pred caminoLibreConBanderitas (t: tablero, p0: pos, p1: pos, b: banderitas) {
  (∃s: seq < pos >)(caminoLibreSinDefinirMinasAdyacentesALaUltimaPosicionconBanderitas(t, p0, p1, s, b) ∧L
  8 > minasAdyacentes(t, p1) ≥ 1)
}

```

```

pred todasLasPosicionesSinMinasAdyacentesConUnaPosicionConfirmadaEnJugadaYCaminoLibreAPertenecenAJugada
(t: tablero, p: posicion, j: jugada, b: banderitas) {
  (∀q: pos)((posicionPerteneceAlTablero(t, q) ∧
  ¬ posicionPerteneceABanderitas(q, b)) ∧L
  (0 = minasAdyacentes(t, q) ∧
  caminoLibreConBanderitas(t, p, q, b) ∧
  existeUnaPosicionDelCaminoLibreQuePerteneceAJugada(t, p, q, j, b))) →L
  posicionPerteneceAJugada(q, j))
}

```

```

pred existeUnaPosicionDelCaminoLibreQuePerteneceAJugada (t: tablero, p: posicion, q: posicion, j: jugadas, b: banderitas) {
  ( $\exists m : pos$ )( $(posicionPerteneceAlTablero(t, m) \wedge$ 
   $posicionPerteneceAJugada(m, j)) \wedge_L$ 
   $(0 = minasAdyacentes(t, m) \wedge$ 
   $caminoLibreConBanderitas(t, m, p, b) \wedge$ 
   $(\exists s : seq\langle pos \rangle)(caminoLibreSinDefinirMinasAdyacentesALaUltimaPosicionConBanderitas(t, q, m, s, b))))$ 
}

```

### 3.9. Ejercicio 9

```

pred hayPatron121 (t: tablero, j: jugadas) {
  ( $\exists s : seq\langle pos \rangle$ )( $esPatron121(t, j, s) \wedge_L$ 
   $(\exists p : pos)(posicionPerteneceAlTablero(p, t) \wedge_L posicionNoPerteneceAJugadas(p, j) \wedge$ 
   $(\forall i : \mathbb{Z})(0 \leq i < |s| \rightarrow_L sonAdyacentesNoDiagonales(p, s[i])))$ 
)
}

pred esPatron121 (t: tablero, j: jugadas, s: seq\langle pos \rangle) {
   $posicionesPerteneceAlTablero(t, s) \wedge posicionesPerteneceAJugadas(j, s) \wedge (|s| = 3) \wedge_L$ 
   $(\exists t : seq\langle pos \rangle)(secuenciaOrdenada(t) \wedge esPermutacion(s, t) \wedge_L cumple121(s));$ 
}

pred posicionNoPerteneceAJugadas (p: pos, j: jugada) {
   $(\forall i : \mathbb{Z})(0 \leq i < |j| \rightarrow_L j[i][0] \neq p)$ 
}

pred sonAdyacentesNoDiagonales ( $p_1, p_2 : pos$ ) {
   $((p_1[0] - p_2[0]) = 1 \wedge (p_1[1] - p_2[1]) = 0) \vee$ 
   $((p_1[0] - p_2[0]) = 0 \wedge (p_1[1] - p_2[1]) = 1) \vee$ 
   $((p_1[0] - p_2[0]) = -1 \wedge (p_1[1] - p_2[1]) = 0) \vee$ 
   $((p_1[0] - p_2[0]) = 0 \wedge (p_1[1] - p_2[1]) = -1)$ 
}

pred posicionesPerteneceAlTablero (t: tablero, s: seq\langle pos \rangle) {
   $(\forall i : \mathbb{Z})(0 \leq i < |s| \rightarrow_L posicionPerteneceAlTablero(s[i], t))$ 
}

pred posicionesPerteneceAJugadas (j: jugadas, s: seq\langle pos \rangle) {
   $(\forall i : \mathbb{Z})(0 \leq i < |s| \rightarrow_L posicionPerteneceAJugadas(s[i], j))$ 
}

pred cumple121 (s: seq\langle pos \rangle) {
   $((minasAdyacentes(s[0]) = 1) \wedge (minasAdyacentes(s[1]) = 2) \wedge (minasAdyacentes(s[2]) = 1))$ 
}

pred secuenciaOrdenada ( $p_1, p_2 : pos, s : seq\langle pos \rangle$ ) {
   $(s[0] = p_1 \wedge s[|s| - 1] = p_2) \wedge$ 
   $(\forall i : \mathbb{Z})(0 \leq i < |s| - 1 \rightarrow_L esAdyacente(p[i], p[i + 1]))$ 
}

```

```

pred esPermutacion (s, t: seq(pos)) {
  (|s| = |t|) ∧
  (∀x : pos)(posicionPerteneceASecuencia(p, s) ⇔ posicionPerteneceASecuencia(p, t))
}

```